

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

61
Claim 1 (original): A method for processing storage data that is to be communicated over a network, comprising:

providing storage data to be transmitted over a network;

serializing the storage data using storage encapsulation protocol headers to generate serialized storage data;

encapsulating the serialized storage data using a simple transport protocol to generate simple transport protocol data segments of the storage data; and

encapsulating each of the simple transport protocol data segments into Ethernet frames.

Claim 2 (original): A method for processing storage data that is to be communicated over a network as recited in claim 1, wherein the serializing of the storage data using storage encapsulation protocol headers to generate serialized storage data includes:

receiving the storage data, the storage data including one or both of commands and data, the commands including write commands, read commands, control commands, and status commands;

selecting portions of the received storage data to be serialized, the selected portions including commands and data; and

appending storage encapsulation protocol headers to each of the selected portions.

Claim 3 (original): A method for processing storage data that is to be communicated over a network as recited in claim 1, wherein the encapsulating of the serialized storage data using a simple transport protocol to generate simple transport protocol data segments of the storage data includes:

selecting portions of the serialized storage data; and

appending simple transport protocol headers to the selected portions to generate the simple transport protocol data segments of the storage data.

Claim 4 (original): A method for processing storage data that is to be communicated over a network as recited in claim 1, wherein the encapsulating of each of the simple transport protocol data segments into Ethernet packets includes:

generating media access controller (MAC) header;

appending the simple transport protocol segments to the MAC header; and


appending a cyclic redundancy check (CRC) to the simple transport protocol segments.

Claim 5 (original): A method for processing storage data that is to be communicated over a network as recited in claim 3, wherein the simple transport protocol headers each include at least a handle field, a type field, a length field, a sequence number field, and an acknowledgment field.

Claim 6 (original): A method for processing storage data that is to be communicated over a network as recited in claim 5, wherein the handle field is used to exchange a handle

during the commencement of a session, the handle being exchanged between a initiator and a target of the network.

Claim 7 (original): A method for processing storage data that is to be communicated over a network as recited in claim 5, wherein the sequence number field is configured to count Ethernet frames.

 Claim 8 (original): A method for processing storage data that is to be communicated over a network as recited in claim 5, wherein the acknowledgment field is used to exchange positive and negative acknowledgments of transactions.

Claim 9 (original): A method for processing storage data that is to be communicated over a network as recited in claim 1, wherein the storage encapsulation protocol contains a tag so that data segments and data segments of the storage data can be matched to a correct command.

Claim 10 (original): A method for processing storage data that is to be communicated over a network as recited in claim 1, wherein the STP transport protocol is configured to provide a stream of bytes arriving in the same order as they were sent.

Claim 11 (original): A method for processing storage data that is to be communicated over a network as recited in claim 1, further comprising:
appending an IP header to each of the simple transport protocol data segments.

Claim 12 (currently amended): A method for processing storage data that is to be communicated over a network as recited in claim 1, wherein the storage data is selected from one of small computer system interface (SCSI) ~~SCSI~~ data, AT Attachment Packet Interface (ATAPI) ~~ATAPI~~ data, and ultra direct memory access (UDMA) ~~UDMA~~ data.

Claim 13 (original): A method for communicating storage data over an Ethernet network using a non-TCP lightweight transport protocol, comprising:

providing data having a peripheral device protocol format, the data to be communicated over the Ethernet network;

selecting portions of the data;

attaching storage encapsulation protocol (SEP) headers to the selected portions of the data;

attaching simple transport protocol (STP) headers to one or more of the selected portions having the SEP headers to produce STP packets; and

encapsulating the STP packets into Ethernet frames for communication over the Ethernet network.

Claim 14 (currently amended): A method for communicating storage data over an Ethernet network using a non-TCP lightweight transport protocol as recited in claim 13, wherein the peripheral device protocol format is one of a small computer system interface (SCSI) ~~SCSI~~ format, an AT Attachment Packet Interface (ATAPI) ~~ATAPI~~ format, and an ultra direct memory access (UDMA) ~~UDMA~~ format.

Claim 15 (original): A method for communicating storage data over an Ethernet network using a non-TCP lightweight transport protocol as recited in claim 13, wherein the STP headers include at least a handle field, a type field, a length field, a sequence number field, and an acknowledgment field.

9
Claim 16 (original): A method for communicating storage data over an Ethernet network using a non-TCP lightweight transport protocol as recited in claim 15, wherein the handle field is used to exchange a handle during the commencement of a session, the handle being exchanged between a initiator and a target of the network.

Claim 17 (original): A method for communicating storage data over an Ethernet network using a non-TCP lightweight transport protocol as recited in claim 15, wherein the sequence number field is configured to count Ethernet frames.

Claim 18 (original): A method for communicating storage data over an Ethernet network using a non-TCP lightweight transport protocol as recited in claim 15, wherein the acknowledgment field is used to exchange positive and negative acknowledgments of transactions.

Claim 19 (original): A method for communicating data over an Ethernet network using a non-TCP lightweight transport protocol, comprising:

providing data having a virtual interface format, the data to be communicated over the Ethernet network;

selecting portions of the data;

attaching simple transport protocol (STP) headers to the selected portions of the data to produce STP packets; and

encapsulating the STP packets into Ethernet frames for communication over the Ethernet network.

Claim 20 (original): A method for communicating data over a network using a non-TCP lightweight transport protocol, comprising:

providing data, the data to be communicated over the network;

selecting portions of the data;

attaching simple transport protocol (STP) headers to the selected portions of the data to produce STP packets; and

encapsulating the STP packets into frames for communication over the network.

Claim 21 (original): A method for communicating data over a network using a non-TCP lightweight transport protocol as recited in claim 20, wherein the data is one of storage data, network data, file data, and virtual interface data.

Claim 22 (original): A method for communicating data over a network using a non-TCP lightweight transport protocol as recited in claim 20, wherein the network is configured to communicate storage data.
